We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

5

Coupling Activity and Performance Management with Mobility in Vehicular Networks

Miguel Almeida[1] and Susana Sargento[2]
[1]*Nokia Siemens Networks, Universidade de Aveiro*
[2]*Instituto de Telecomunicações, Universidade de Aveiro*
Portugal

1. Introduction

We live in a mobile, fast paced world, where users are constantly on the move. Transportation plays a major role in this matter. Users thrive for services being promptly delivered anytime and anywhere. Nevertheless, business models still focus around Content Service Providers (CSP) and Network Service Providers (NSP), who, as trusted entities, provide more than the connection, further focusing, as time evolves, on the service delivery capitalization. As the trends start to position these providers as the relay points for the information to be conveyed into 3rd party cloud services, the delegation of management functions is also outsourced to the 3rd party entities. It is in this view that the remote management of vehicles becomes of the utmost importance, since connectivity allows the delivery of novel services built around the monitoring of the vehicles' conditions, location and user preferences. The immediate benefits would result in presence/location awareness for retrieval of additional information of the surroundings, or even mechanical support, mechanical failure prediction or detection, based on the continuous monitoring of the vehicles hardware sensors, as well as a whole plethora of new advantages, propelled by the collection of performance and behavior information.

Vehicular networks are inherently associated with high mobility scenarios and this fact introduces new requirements. Usually associated with high velocity patterns, the requirements to support these networks are mainly positioned around the enabling of fast mobility management protocols, and hence interfaces, gifted with the extensibility potential for the exchange of additional information. Furthermore, when considering vehicular scenarios, network mobility and efficiency are two crucial features which need to be kept in mind at all times. They have special influence over the choice of the protocol used to gather information from the vehicles towards the network. These requirements lead us to consider a framework that was originally designed for the management of the mobility of the terminals, and which therefore supports mobility with a high efficiency ratio in terms of resource consumption. This framework, the IEEE 802.21 Media Independent Handovers (802.21-2008, 2009), contains functionalities and elements that can be extended with advanced reporting capabilities to provide seamless reporting in heterogeneous technologies and environments. Using IEEE 802.21, it is also possible to integrate the actions of reporting with the actions of network decisions enforcement. We show that this approach provides a significant set of functionalities not achieved with current approaches, while reducing the overhead on cross-layer reporting. The typical approach is to perform such procedures above the IP layer.

Besides reducing the overhead, gathering performance and action reports at lower layers also saves on signaling and simplifies the protocol stack. When bringing mobility into the picture, these concerns become even more crucial.

Knowing that different types of devices have different groups of requirements in terms of network, hardware and applicational capabilities, the primitives with which all of them interface should be the same: a common Application Programming Interface (API) which is mobility driven and that cleanly exposes management functions (already under evaluation in current research) for seamless mobility and reporting. Management frameworks today also introduce, as a requirement, the definition of interfaces to 3rd party entities. We consider the central management entity to be a cloud of functionalities and of centralized intelligence, which allows interfaces for other cloud services, thus empowering the CSPs with new advanced services and new ways of capitalizing the management functions. By considering the several existing approaches, we derive a solution which combines the most commonly used web services in order to provide a Cloud view of the performance of the several vehicles.

In this chapter we present a solution to collect performance and behavior related information from different communication layers of the vehicles, while keeping in mind the major requirements associated with the inherent properties of the technology which will be discussed along with relevant use cases. Performance management represents a topic which is not widely covered when dealing with vehicular networks, and very little information can be found regarding this subject in the literature. Most research is being conducted in topics related with Vehicular networks focus on mobility management and communication techniques. It is our main goal to evaluate the performance penalties introduced by several layers: hardware, network, session and application. Service performance can be evaluated at any layer without depending on a specific technology, while enabling media independent service reports. It is in this context that vehicles, connected via multiple network access technologies, will report user activities (user behavior related), performance metrics of the mechanical hardware, of the network and of the applications running.

The chapter is organized as follows. Section 2 describes the relevant approaches to deal with the collection of performance information and user behavior activities. Section 3 presents the architecture and the main functional entities to support the mobile user vehicle reports integrated with network reconfiguration trigering. Section 4 depicts the performance comparison of the reporting approach against existent mechanisms, both on a qualitative and quantitave basis. Finally, section 5 presents the most important conclusions from the chapter.

2. Background

This section details the relevant approaches to deal with the collection of performance information and user behavior activities. It details several means to gather information and to present it in a cloud oriented solution (Voas & Zhang, 2009). While considering that vehicles are moving and exposed to different environments, different contexts and different conditions, information can be extracted and conveyed into a platform which, Extracts, Transforms and Loads (ETL), processes it according to predefined metrics, or Key Performance Indicators (KPI), and allows the management parties to evaluate the performance and to take actions. The proposals presented bellow represent efforts in trying to bring the devices closer to the cloud in terms of performance management features. The following subsections detail the technologies which are employed (Section 2.1), compare and contextualize the approaches

to provide a clear view of their adequacy in terms of offered features and usage drawbacks (Section 2.2), and present the current reporting architecture and KPIs (Section 2.3).

There is already some work related to the interconnection of the devices with cloud services for monitoring and management purposes. These studies are, however, not extensive. Most of the efforts were related to sensors as the main analysis use cases, which are not necessarily mobile. Mobile devices introduce additional concerns, since mobility requires maintaining connectivity upon movement. Even in those scenarios, management related issues are still little explored, as stated in Gurgen & Honiden (2009). In Gurgen & Honiden (2009), the authors provide the major requirements for the definition of a platform to manage such devices. In Jung et al. (2007), the work is more focused on a security aware, technology agnostic framework, using Simple Network Management Protocol (SNMP) (J. Case, 1990) to gather information into a command center. This last work is more connected with our proposal, but our list of requirements goes beyond security, having mobility on the top of the list.

Another major concern is the management of the devices in the Cloud, i.e., in an online distributed tool, which appears to the end users as a centralized Graphical User Interface (GUI). This vision on the management of devices is more related to the concept of the Internet of Things (IoT), in which each vehicle can be seen as a thing. In Mohinisudhan et al. (2006), SNMP is used to incorporate hybrid automobiles with a performance monitoring system. Johansson et al. (2005) underlines the usage of a Controller Area Network (CAN) in the automotive market. CAN is a serial bus communications protocol with the purpose of interconnecting sensors, actuators, controllers and other elements. It defines the physical and data link layers for an efficient and reliable communication between the entities. In Johansson et al. (2005) it is presented an integration example with a passenger car, a truck, a navy boat and a spacecraft. In this work the authors also describe the concept of CAN gateways, which provide a way to integrate CAN-based networks with other networks and protocols. This approach is useful in the context of coupling vehicular devices with a performance management platform, since it allows the integration of industry deployed lower layer mechanisms (very oriented to specific vehicle parts' sensors) with network management solutions such as the one here presented.

Extensible Messaging and Presence Protocol (XMPP) (Peter Saint-Andre, 2009) was created for user communication purposes and has already been used for device integration with the cloud, even if only as a protocol capable of interconnecting sensors in an asynchronous wireless environment (Hornsby et al., 2009) (without yet being used for the IoT potential it carries). More recent work, (namely Miguel Almeida (2010a)), takes into account requirements for remote management and its procedures. This approach will be further evaluated in the upcoming sections, since it employs a mechanism to easily integrate devices into the cloud. Although Miguel Almeida (2010b) work does not focus on mobility, since it is merely the definition of a framework and of the required extensions to support enhanced reporting capabilities, it defines extensions and allows us to use them to couple reporting and vehicular device management along with mobility. It takes a more lower layer approach to deal with the problem we are solving and, because of that, it will also be detailed in the sections bellow. Next we detail the technological solutions that are used in the aforementioned proposals.

2.1 Description of the involved technologies

In this subsection we describe the technologies involved in the process of collection information from devices, first detailing an array of data collection mechanisms which are considered the most relevant. Then, we present the trend in the protocols used in the web environments.

2.1.1 Data collection mechanisms

Simple Network Management Protocol (SNMP) (J. Case, 1990) is one of the most relevant data collection mechanisms with wider acceptance, and which is represented by a large scale adoption in a multitude of scenarios. In SNMP the information is collected from the agents in the managed devices according to the meta-data detailed in the Management Information Bases (MIBs), from where the information can be polled. MIBs group parameters that are accessible via SNMP. The SNMP agents and stations use a request/reply protocol to communicate which supports standard messages (Get-Request, Get-Response, Get-Next-Request, Set-Request and Trap). The SNMP station uses Get-Request to solicit information from the SNMP agent, which answers with a Get-Response message. SNMP has been evolving over time with increased security and efficiency. Also, one important aspect is the addition of an unsolicited mechanism via Traps. SNMP-Trap is an unsolicited message sent by SNMP agents to the manager. These messages inform about the occurrence of a specific event, and can be used to inform that a link is down or that the agent is reinitializing itself. Traps allow for reactiveness and simplify scenarios where polling is not the best option. Remote Monitoring (RMON) (Waldbusser, 1995) extends this concept by introducing probes and, instead of measuring Network Elements (NE), it focuses more on the analysis of traffic flows. This approach is particularly useful for the identification of third party services or servers, troubleshooting the network problems, security breaches or simply keeping logs of user activities for accounting or profiling.

The Common Management Information Protocol (CMIP) (J. Case, 1990) provides a complete network management framework over many, diverse network machines and computer architectures. CMIP's mode of operation differs from SNMP's, in the sense that the latest was designed for simplicity and ease of implementation. Besides the same functions provided by SNMP, CMIP contains more functionalities, thus allowing a wider range of operation sets. In this framework, any relevant information can be requested from the managed object and can be interpreted according to the managing system. A main drawback of CMIP is its complexity, and therefore, its adoption did not fall in the networking enviornment.

Call Detailed Records (CDR) (Breda & Mendes, 2006) include information of the call duration and failure causes, and are generally used with some lightweight data mining processes to withdraw immediate conclusions. They are largely used by cellular operators to perform some minimal profiling computation in their business intelligence solutions. CDRs are typically generated on a per-call basis: each call can originate a CDR. Although originally the CDR was designed to describe call details for billing purposes, it can be used to trace the call at the business level and retrieve service assurance relevant information. This information complements the Performance Management information by extending the network behavior analysis to the service/subscriber scope, providing the ability to propose new analysis scenarios (e.g. to assess if network is accurate in the service delivery, or which services are more suitable for that network considering the traffic model and user behavior). In order to support our requirements of supporting seamless reporting through inter-technology

environments, CDRs would need to be extensively expanded leading to a high increase in the overhead. The three most common procedures to collect CDR comprise: (1) the real-time transfer of a single CDR each time a call occurs; (2) the near-to-real-time transfer which takes place after several CDRs have been grouped into a single Blocked Generation Log and subsequently sent via Event Forwarding Discriminator (EFD); (3) and the collection of CDR records being stored in File Generating Log.

Other approaches, such as Mobile QoS Agents (Soldani, 2006), installed in the mobile devices, are also being implemented to gather end-user related information. The agents are executed remotely and gather a limited set of performance metrics that provide information to derive the Quality of Experience (QoE), as they are physically near the users. Other proprietary protocols over IP are also being implemented, where the data structure is XML, and the meta-data is defined by the hardware manufactures and interpreted by the performance monitoring solutions with knowledge on the specifications. The agents can gather very specific metrics depending on the device or on the analysis use case, and thus these solutions should be seen as very implementation dependent and customizable. Their usage can be applied to functions such as measure the user feedback from an application (via a pop up questionnaire), substitute road-tests by measuring signal power and SNR to evaluate coverage, analyze network metrics (throughput, delay, bit rates over time), estimate location and deduce trends.

The IEEE 802.21 Media Independent Handovers (MIH) framework deals with the exchange of information for mobility management in heterogeneous environments. This information includes: events, which typically deal with the changes on the link layer level and which may prompt for handover; commands, which serve the management purpose by indicating control information about handovers; and information messages, which provide details on the status of the extended services of the network and information on the available networks.

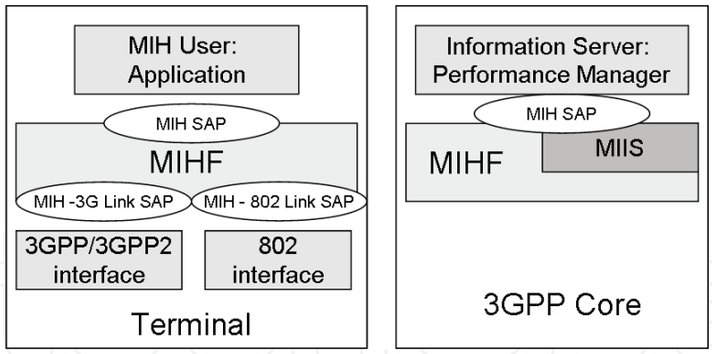


Fig. 1. IS Server Location

Media Independent Information Services (MIIS) were defined to support various Information Elements (IEs), which can be used to provide further information for handover decisions. Fig. 1 shows how the MIH Users can interact with the MIH Function (MIHF), and how the Service Access Points (SAPs) are implemented in order to allow communication with the lower layers.

2.1.2 Integrating with the webcloud

Up to now, the presented approaches are focused on ways to exchange information using existent technologies that were primarily created for that particular purpose. The requirements now invoke wider concerns, namely those related with the gathering of

information into a cloud of services. In this sense, from lower layer related protocols we now move into solutions which were defined to integrate computers connected to the Internet, which is the de-facto environment for the cloud based applications, and the substratum for all presentation layers.

The typical model for communicating with the cloud, or web service guideline, is based on Representational State Transfer (REST) interfaces over HTTP (Fielding, 2000). REST provides a clear interaction model that enables a powerful and flexible solution through simple interfaces in a scalable environment. REST and Simple Object Access Protocol (SOAP) (Don Box, 2000) are associated to HTTP as means to convey information understandable by the cloud. SOAP provides the support of objects over HTTP; however, SOAP faces a scalability issue because it usually requires a large amount of technology to establish bidirectional invocation: it usually requires an HTTP web server, coupled with an application server to enable the web service environment. Moreover, current trends resort to REST over HTTP due to its simplicity and ease of usage given the mapping with of the HTTP methods (GET, PUT, DELETE and POST). It also provides a long-lasting interface that is not coupled with the business logic behind the interface. When deploying these mechanisms, manufacturers are concerned about assuring a future proof solution, and hence look at the choices which grant them a more secure bet on the long term.

XMPP (P. Saint-Andre, 2004) offers good conditions as a transport protocol for applications within the web services' (WS) scope since it offers reliability, synchronous and asynchronous delivery of messages, and does not require a complex set of features such as WS-Routing and WS-Referral to ensure identity trace back (Fabio Forno, 2005) within private domains, since addressing is not only IP based. XMPP was conceived as an alternative Instant Messaging protocol, but has been evolving to a broader concept. Given the fact that it is open and XML based, it became easy extensible and became an IETF standard.

2.2 Taking advantage of the existing approaches

In this section we evaluate the several approaches to deal with the problem of collecting performance management and behavior related data, and presenting it in the cloud - a place which is spatially and software distributed, but which creates an abstraction to present a centralized logic to the users accessing it. Users access a GUI which hides all the hardware and software complexity behind it. Bellow are two major contributions that provide an answer to this problem and which will be evaluated and used. The first handles performance collection on higher layers and takes advantage of existing solutions at the applicational layer, namely using web oriented protocols, that are user oriented. The second provides a MAC layer solution for the gathering of information using a protocol which was originally conceived for the management of the devices' mobility. We will take advantage of both solutions as inputs for the definition of the architecture presented in this chapter. The way both are integrated is explained in Chapter 3.

2.2.1 Using XMPP and REST

As stated, one good approach for collection of performance related information is to perform it in a web oriented environment. Using XMPP and REST (Miguel Almeida, 2010a) brings advantages in terms of collecting user information. According to (P. Saint-Andre, 2004), there are three Core Stanza types defined by XMPP: The <message/>, <presence/> and <iq/>. The first works as a push mechanism to immediately send messages if the destination is online.

Presence relies on publish-subscribe mechanisms through which nodes inform the server of their availability (e.g.: online, away, do not disturb), and is usually distributed among the other nodes in the roster. The last one is a stanza responsible for entities making requests and receiving responses (hence Info/Query) from each other for management, feature negotiation and remote procedure call invocation.

One of the biggest advantages of XMPP is the fact that the addresses can be associated with people or devices such as computers, mobile phones, sensors, routers or cellular network elements (3GPP RAN and Core Network Elements). This is achieved by the use of a Jabber ID (JID), a uniquely addressable ID, which is a valid Uniform Resource Indicator (URI) (Berners-Lee & Masinter, 1998), created according to the following format: person@domain/resource, where person usually represents the user entity; domain represents the network gateway or "primary" server to which other entities connect for XML routing and data management capabilities; and resource, which is of special interest since it allows to identify a specific device associated with the person. Security can be achieved by using Transport Layer Security (TLS) for channel encryption, while authentication is achieved through Simple Authentication and Security Layer (SASL). Regarding the portability and interoperability requirements, XMPP uses the "over-IP" approach and allows the binding of resources to streams for network-addressing purposes. This feature also allows to perform Identity Management via the relationships of the user and of the resource.

One of the requirement of our vehicle scenario is the communication across multiple domains (e.g. across two operators). XMPP (P. Saint-Andre, 2008) allows multi domain management that can be achieved while making use of server-to-server communication. It also allows the capabilities' exchange and location awareness features via the presence stanzas. Regarding the efficiency of the protocol, several activities are being conducted to improve XMPP performance, namely new lightweight version such as (Hornsby & Bail, 2009); however, the major performance issues drive from the presence signaling which can be optimized. This concern can be overcome with proposals like SOAP over XMPP (Fabio Forno, 2005), that would even enrich the performance concerns, since XMPP and SOAP are two XML based protocols, running one on top of the other.

2.2.2 Using media independent handovers

(Miguel Almeida, 2010b) focus on the possibility to merge reporting with mobility intrinsic protocols. Since IEEE 802.21 was developed and is used to provide a lower layer communication framework to deal with the exchange of information in heterogeneous environments, our aim is to further extend it to enable the exchange of end user reports independent from the underlying technologies. Moreover, this extension will allow the seamless integration and activation of network reconfiguration procedures.

By extending the IEEE 802.21 MIIS, end user reporting can be performed at lower layers, using one single protocol to carry all user, vehicle and network related information, which will increase the efficiency of resource consumption. The MIIS is expected to provide mainly static information but, for the envisioned approach, real-time and dynamic information is required. The IEEE 802.21 standard also mentions that dynamic information such as available resource levels, state parameters and dynamic statistics, can be obtained directly from the respective access networks. However, this information usually does not provide a clear view on the end-to-end service performance. Also, the gathering of user behavior related information from the network requires a means to access this information: this can be supported through

the IEEE 802.21, by loosening the concept of the MIIS and supporting new features and functionalities.
To determine the QoS and QoE, it is required to assess the impact of the lower layer information on higher layers at the core side. This information can be related via the cross relation of PoA (Points of Access) with the terminal identification via the SAP (Service Access Points). Therefore, it allows the collection of most of the information locally (either from lower or higher layers), pre-evaluate it and then send it to the network. This view is aligned with IEEE 802.21 which, through the MIIS, can provide an indication of higher layer services supported by different access networks and other relevant information that may aid in making handover decisions. Such information may not be available (or could not be made available) directly from MAC/PHY layers of specific access.
Finally, the support of user and device (vehicular) reports through IEEE 802.21 allows the seamless activation of network reconfiguration procedures, such as session and terminals handover to networks that better match the user/device requirements, to jointly optimize network resources and user experience. In sections 3.1 - 3.3, we further detail and propose extensions to the MIIS to support a more detailed communication of application level parameters, and introduce more intelligence upon handover decision.

2.3 Performance and behavior management

As defined by ITU-T, the Telecommunications Management Network model (TMN) (ITU, 1996), used for managing open systems in a communications network, establishes four management layers comprising: (1) the element management, which entities are hierarchically above and gather the information which is collected by each Network Element; (2) the Network management system, which evaluates these metrics (after a Transform and Load process); (3) the Service Management, which is in charge of taking into account the previous layer and extrapolate conclusions that can lead to active changes in the network; (4) and the business management layer, which introduces the agreement levels that need to be accomplished. The Network Elements (NE) are typically the network nodes which interact with the delivery systems. Operations, Administration and Maintenance (OAM) describes a set of management levels and their interactions. The concept has more recently evolved to include Provisioning and Troubleshooting. It ideally would imply the cross view of the TMN model with the Fault, Configuration, Accounting, and Performance and Security (FCAPS) functionalities.
To provide a clear view on the performance of the vehicles, indicators need be defined according to the relevant metrics in the vehicle network. Key Performance Indicators (KPIs) are a set of selected indicators used for measuring the current performance and trends. KPIs highlight the key factors of the current performance and warn of potential problems. Considering a counter as the most elementary value which is collected from a vehicle, a KPI can simply be equal to a counter or to an arithmetic abstraction of counters that can be applied to monitor a certain part of the network, functionality or protocol. KPIs play a major role in creating immediate and relevant feedback on the performance of a certain element (may it be network, hardware, or behavior).

2.3.1 Generic reporting tool architecture

Since we are proposing a remote management platform, the whole system would not be complete without the inclusion of an architecture to evaluate the performance of the vehicles

in the cloud. This Reporting Tool receives the information from the devices and allows an online verification of their performance by the end users. Fig. 2 shows the main components which are typically included in a generic architecture for a reporting tool. Bellow we explain the major functionalities of each component and their relevance to the architecture. The Reporting Engine is the mind behind the Reporting Tool (Fig. 2). It is responsible for the database queries, it processes the results and displays them in a defined format. It provides all the data visualization capabilities, offering different pre-defined models and allowing the user to create their own. These pre-defined visualization models are important, because they allow the manipulation of data in different dimensions, providing different reports for different types of end users, and even for different type of analysis, starting from a unique data set. Related to these models, there is an important reporting component, which is the KPI set. KPIs are defined in configuration components and can be either calculated on the fly by reporting engine, or pre-calculated and stored in the Reporter Database. The Automated Knowledge Discovery model is another important part of this reporting engine, and provides the very important feature of automatic data monitoring, searching for patterns in the network behavior for the purpose of forecasting upcoming events, such as the Operation, Maintenance and Optimization needs.

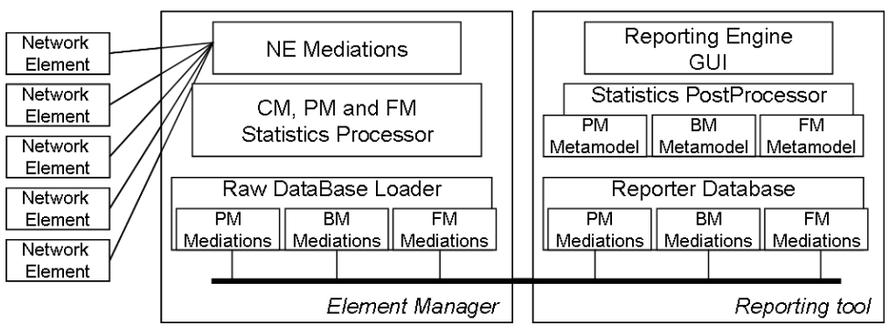


Fig. 2. Generic Reporting Tool Architecture

The Reporter Database is a data warehouse designed for the coherent integration of diverse data sources, dimensioned to optimize the data discovery and reporting. This data repository modulates all the network topology into a hierarchal object structure, which provides the capability to analyze the entire network. This analysis can focus on the correlation of different parameters that can be Configuration, Performance or Fault Management related. A possible use case would be to assess what kind of configuration optimizes better the performance of the network, by improving network capacity and reducing its faults. This analysis can be extended in time and from different network perspectives, as historical and object data aggregation is possible. Moreover, the database primitives allow for the storage management and provide all the data access information to other layers. This database is modulated based on NE specific metadata, which defines the Object Class (OC) structure, and for each OC all the PM Measurements and related list of PM Counters and PM counters aggregation rules. The FM metadata is generic for all the OC, defining a list of failures that can occur.

The Statistics Post-Processor is a software component that plays a decisive part on the reporting process. It is responsible for the entire object and time aggregations, which enhances the analysis capabilities, allowing the time trend analysis and drilling through the network objects, enabling a great diversity of network analysis. The aggregation rules are all defined through metadata specific for each NE, and provide information on how PM counters must be

aggregated. The statistics Processor is responsible for converting all the diverse data gathered from the NEs according to a structured and generic meta-model. This particular module processes Configuration, Performance and Failure Management Information.
The Raw Database loader is responsible for providing interfaces for the access relating to data storage management features. It is another function of an ETL procedure which uploads the gathered information into the raw databases. This module includes interfaces for mediation of the interactions between the EM and the analysis tools that evaluate the collected data. These interfaces answer to the Reporting Tool for requests related to Performance Management (PM), Configuration Management (CM) and Fault Management (FM) data.
The NE Mediations manage the interactions between the Element Manager Module and the several Network Elements in the network. They are responsible for the collection of the Performance and Fault Management functions existing in each of the elements of the network. The NE Mediation Module implements the Extraction part of an ETL procedure. Each network element monitors its performance through the Performance Mediation. A subset of that module is responsible for the communication with the Element Manager. That interface is divided into three types of primitives relating to the type of data which is to be transported: PM, CM and FM. The first presents metrics related with the continuous operation of the equipment, while the second indicates the configuration setup, including information such as topology and capabilities. FM is a more urgent type of data as it indicates critical issues to be evaluated.

2.3.2 Types of metadata

As stated, the main objective of this chapter is to focus on the end-to-end reporting capabilities between the devices and the cloud, while providing mechanisms and information so that decisions can be made and measures can be taken if problems occur. However, the decision making and acting components are not discussed. When considering reporting functionalities, the typical supported metadata types are: CM, PM and FM. The work presented in this chapter also considers Behavior Management (BM), in the sense that it allows the gathering of metrics associated with the behavior of inhabitants of the vehicles and their interactions with the vehicles.
Configuration Management metadata is responsible for the mapping between the different NEs present in the network and their components into a coherent and structured Object Class model. This way, CM metadata is used to identify objects with the same properties and to maintain possible occurrences of an object in the object class hierarchy. Two types of objects can de defined for this model, Managed Objects (MO) and Reference Objects (RO). Managed Objects refers to objects that are directly related elements present in the network that can be managed, configured, manipulated, which are obviously the NE elements and their components e.g. a Node B and its Cells. Reference Objects refers to virtual reporting dimensions, i.e. elements that are virtually created to ease the network analysis by dividing and grouping the network into smaller segments thus reducing the analysis complexity. This Reference Objects are created and stored in the Reporter Database by the Statistics PostProcessor module, using the CM metadata information.
Performance Measurement metadata is responsible for defining, for each network element, all the PM measurements and Counters and relating them to the CM data, i.e. to the OC structure. As network element represents a specific role in the network, there will be a different set of measurements/counters for each NE. The number of measurements and counters needed

to monitor a specific NE is dependent on the NE complexity, ranging with the number of functionalities. A PM measurement is a logic representation of a NE functionality that defines a set of counters that monitors the network performance behaviour. A PM counter is the fundamental element of the performance monitoring process, as it provides detailed information ranging from specific procedures up to group functions. As counters are the basis of PM, they are used to develop different kinds of aggregations such as KPIs and Reports. This way, different kind of users and analysis can be satisfied with only single tool.
Fault Management metadata defines the mapping between all the NE components and the fault events that describe system failures, either hardware or software driven. FM metadata thus relates OC with incoming network failure notifications. These failures are categorized and ranked by severity, which can range from debug to emergency state. The special characteristic of this type of data is the fact that it typically has an unsolicited behavior and requires near real time functionalities.

3. Architecture description

The following section depicts the architecture and the main functional entities that need to be included to sustain the previously defined requirements, namely, the inter-technology scenarios and the support of end user terminal reports integrated with network reconfiguration triggering. Fig. 3 presents the vision explained in this chapter. Vehicles are moving freely and through a wireless connection, which can be WiFi, WiMAX or 3GPP based (UMTS, iHSPA or LTE), and are connected to a CSP. By using a mobility management protocol like the Media Independent Handover with extended reporting capabilities, the performance measurements of the vehicles and the behavior of the users can be gathered, stored and evaluated within a cloud. This allows early problem detection, location and context awareness, remote assistance, and a plethora of services from which fleet management functionalities should be underlined.

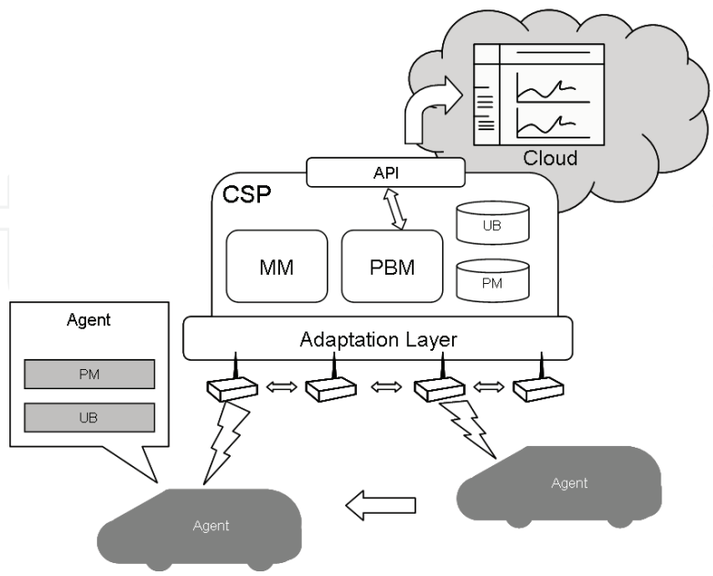


Fig. 3. Interactions between the vehicles and the cloud

Each vehicle has an agent installed which sends information to the cloud, and is handled by entities connected to a logic bus, the Media Independent Handover Function (MIHF). On the other side, the Performance and Behavior Management module (PBM), collects that information and stores it according to the type of data (User Behavior (UB) or Performance Management (PM), which will be detailed in the next subsection). 3rd Party Cloud services access this information and apply analysis algorithms (data mining procedures can be applied but are not within the scope of this work), to present graphics explaining occurrences of problems in certain vehicle models or in certain zones.

3.1 Architecture specification

Fig. 4 shows the main required entities for the proposed reporting architecture. End user Behavior reports are communicated by the Multimedia Application to the Behavior Manager at an agent (BM@Agent) installed in the vehicle. The Performance Manager (PM@Agent) is a MIH User as well, and collects information from both the lower layers (QoS information) and upper layers (QoE related information).The PM also interacts with the running applications and the vehicle's mechanical parts as well as the software/firmware for monitoring purposes. The agent is a MIH entity that is responsible for gathering information and communicating performance and behavior metrics. Lower layers report metrics that are extracted from the technology drivers, including link and network layer values, such as throughput, bit rate and SNR. From the upper layers, the PM will receive the information regarding service performance, mainly related with end-to-end performance and QoE feedback. To achieve this, these modules have open interfaces, which can be used through specified primitives, as will be explained in the next subsection. Lower layer information can also be retrieved directly by the Media Independent Information Service (MIIS). The MIIS (see Fig. 1) collects the data on the network side and feeds the relevant user behavior and performance values to the Performance and Behavior Manager (PBM@network).

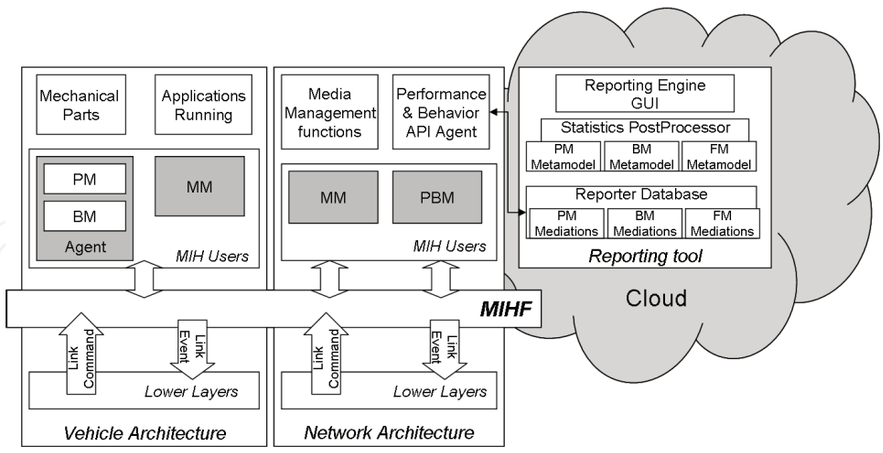


Fig. 4. Interactions between the vehicles and the cloud

The PBM@network is responsible for the interaction with the multiple terminals to perform profiling analysis for individual and group behaviors. It comprises a database and interacts with the API Management Agent, which is responsible for allowing access to the reporting tool and 3rd party services. The typical approach to evaluate a user's opinion on a service, and hence depict his profile, is to collect end user reports after a service has been delivered, and

converge the opinion with the provided service's characteristics. However, other methods can provide an equally efficient evaluation of the user's acceptance to performance trade offs. The QoE and expected experience form a couple of properties that cannot be considered separately. A user may be willing to accept lower performances if the contract fee is lighter. This conclusion and consequent profiling can be drawn from the user behavior. After applying the profiling analysis algorithm, the PBM formats the information to feed the Mobility Manager (MM) and stores the results.

The MM receives the inputs from terminals and decides if an action is required. The MM can use this input to take decisions, activating events in the terminal or events in the network for optimization purposes. This process will make use and extend the IEEE 802.21 signaling. Other proposals (Chung et al., 2008), (Jesus et al., 2007) deal with the mechanisms involving mobility decisions and mobility signaling more deeply. To better understand this process, the core network should be seen as a mediator of information. The vehicles will send information to the CSP infrastructure to be handled by the PBM, and this information will be made available to the in-cloud applications through the Performance and Behavior API Agent (see subsection 3.4). It is also through that agent that the in-cloud applications can interact with the vehicles. Fig. 4 shows an application in the cloud performing the remote management of the performance of the devices. This scenario shows how the mediated data collected from the devices can be outsourced to other services.

3.2 Signaling

To better understand the message flow, we will consider the scenario where a user contains a multi-homed terminal connected to two wireless networks (e.g. WiFi and UMTS), but is using the WiFi one (Fig. 5). Periodically, the multimedia applications report activity updates and performance metrics. These messages are not IEEE 802.21 messages (Action messages in Fig. 5), but are internal primitives. The same application will periodically issue another message (Performance Report) informing about the relevant performance metrics for that particular service. As shown in Fig. 5, an Action message is sent from the user application to the BM@agent indicating that the user wants to receive a video and has issued for a VoD request. The message should be according to the type: Action (Application ID, Type of Request, Timestamp), thus specifying the application type which is being used (Video, Audio, Gaming, Browsing, etc.) and the type of request (VoD, Streaming, Conference, Starting Game, etc.). Following that procedure, different Performance messages will also be sent regularly. The application issues a message containing the following structure: Performance (SourceID, MetricType, Value, Timestamp), thus depicting the type of metric being reported and the value for that metric. These messages are issued locally and then mapped to IEEE 802.21 by the MIH Users (both BM and PM) for transmission to the network side (in the form of the messages and procedures depicted in Section 3.3. Moreover, the PM@Agent receives Performance updates from the hardware adaptation as explained in Section 3.3.2.

Ideally, the agent@terminal has well defined interfaces with the applications, and each application can be responsible for reporting the user's activities and performance feedback. However, we introduce these entities as functional blocks for a better comprehension and easier compliance with legacy applications. Hence, both message types Performance and Action do not reflect any type of signaling protocol but are instead internal primitives. The BM and the PM are now ready to report this information to the network using the MIH Function. When the Information Service requests for a UE update (MIH_Get_Info.request), it gets a

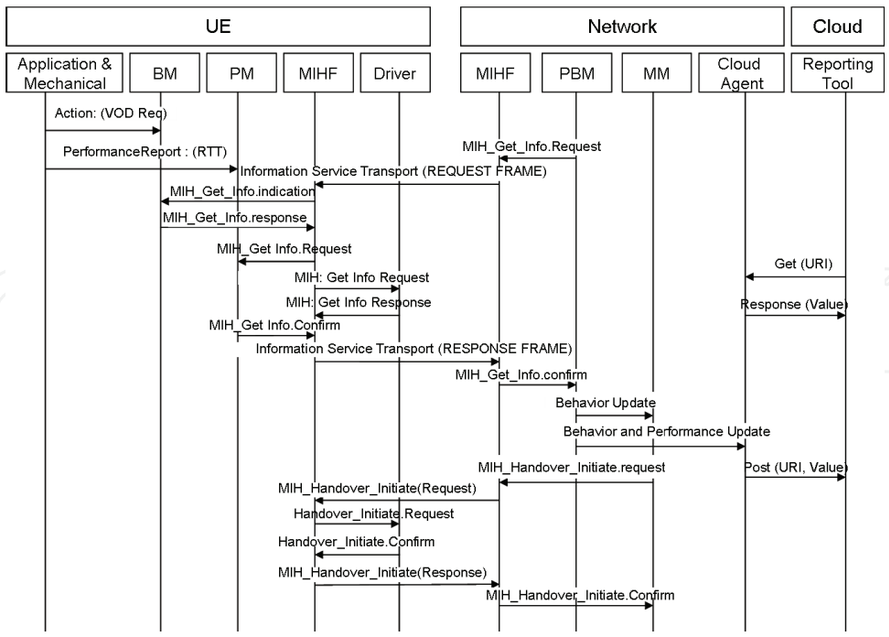


Fig. 5. Signaling diagram. URI is constructed from through the vehicle ID as explained in Fig. 7. (*Mechanical and Application Adaptations which are the Agent interfaces)

response containing the QoS performance from the application, from the lower layers, and also the reported user action (via the Information Service Transport message). The IS receives an update on the user status, and forwards this information to the PBM which evaluates the QoS parameters, increments the user profile and communicates the changes to the MM. The MM will evaluate the feasibility of this network for the desired service. Since it already knows that the terminal has another interface with different properties (via standard IEEE 802.21 signaling: link_up message), it decides that a link with more bit rate would be better for the services in use. It then issues a handover request to the terminal, so that it performs a handover to LTE.

Whenever an action is taken by a user, the system needs to identify if the user is satisfied with the current quality of the service (according to QoE parameters): the desired characteristics and the used application's requirements should be taken into account to assess if they are being met. If not, a change is required, by using another available interface (performing a session handover) or switching to another PoA in order to enhance the terminal reception conditions. This makes it possible to optimize the network or re-allocate users on different PoAs. When Behavior and Performance Updates are collected by the PBM@Network, the Cloud Agent is notified. It should then lookup the 3rd party entities which have interest in receiving this update and use the POST method to transfer that information into the destination web applications. This information can also be requested from the 3rd Party services (which we designate as cloud in Fig. 5), using the Get method. Both GET and POST methods use the URI, which is formulated via the identification of a user and the element of a vehicle belonging to that user as well as the metric which is to be retrieved (this mapping is done via the SAP ID and the metric type as explained n Section 3.4). To support this view, it is required that both the Cloud Agent and the 3rd party entities (in Fig 5 the web service

is exemplified by a performance reporting tool) are running a web server, since the REST methods are used via HTTP.
The way the mapping of web resources is done is detailed in subsection 3.4. In this section, we demonstrate the signaling flow to better explain how the information is conveyed to and from a web cloud based application. In general terms, the main concern is to cache the information locally and send an update to the web applications with which the CSP has an SLA, and which has expressed intention in receiving a particular device's instruction. It is assumed that this process is preceded by a pre-enrollment phase, by means of which the owner of the vehicle agrees on the availability of his data, as well as remote management functionalities being sent to the vehicle. The defined signaling will allow both synchronous and asynchronous reporting messages that are of relevance in order to support two types of information: a) periodic performance or behavior data that has no crucial impact on the performance of the vehicle; b) relevant and urgent information that might indicate a malfunction or a possible problem and thus should be sent in an unsolicited way. Although the use case of sending a command from the cloud is not expressed in Fig. 4, it is supported as explained in subsection 3.4, via the usage of a Rest command which is received at the API on a particular resource which unequivocally identifies the vehicle and the command to be remotely executed. The API agent then conveys that command to the destination vehicle through the UBM.

3.3 Information elements

The IEEE 802.21 already defines general IEs, access network specific IEs, PoA specific IEs and other IEs. Information Service elements are grouped into three categories: a) General Information and Access Network Specific Information, which give an overview of the different networks; b) PoA Specific Information that provides information about different PoAs for each available access network; c) Other information that is access network specific, service specific, or vendor/network specific. Next, we propose to include the Service Performance IEs and the User Behavior IEs to be used in the previously presented architecture, thus extending the ones defined in the standard, while taking (Miguel Almeida, 2010b). The agent at the vehicles handles the following 3 types of messages:

- Action (SourceID, Type of Request, Timestamp)
- Performance (SourceID, MetricType, Value, Timestamp)
- Alarm (SourceID, MetricType, Value, Timestamp)

The first two can be easily handled by normal MIIS procedures, while the last one, given its nature, would benefit from a more unsolicited behavior. One way to solve this problem is to use the MIH_Get_Information.request message carrying data that would indicate the occurrence of an alarm situation. The MIH_Get_Information.request is sent to the MIHF in the termina,l and then in the network side, a MIH_Get_Information.indication notifies the PBM@network (corresponding MIH User).
The MIH_Get_Information.response brings the confirmation that the alarm has been received. This provides a good workaround for the lack of unsolicited messages between MIH Users. The Alarm reflects the over crossing of a threshold value; the format of the message is the same as the one of the Performance, but only indicates the urgency of the problem to the network. For the purpose of supporting interfaces with the Controller Area Network (CAN) existing in the vehicles (Johansson et al., 2005), the agent should also be able to act as a CAN gateway. Since CAN and IEEE 802.21 are both lower layer based approaches, the overhead is minimal,

thus presenting a resource efficient solution. The way our framework handles this integration is explained bellow in Section 3.3.2.

3.3.1 User behavior information elements

The type of active applications is usually communicated in the bootstrap of an application. We define it to be in the form TYPE_IE_UB_ACTIVE_APP_ID. It contains an index of the application regarding the content a user is requesting. After reporting the active application, several requests can be made by the user. The message type that should be used for this type of report will be: TYPE_IE_UB_ACTION (ApplicationID, UserAction, Timestamp). The User Action field is defined in Table 1, which contains information on the actions that are required to be supported for the previously mentioned services. The Timestamp is a time reference.

ApplicationOn	ApplicationOff	RequestChannel	IdleMode
EndConversation	InitiateConversation	RequestURL	ActionMode
SendMessage	ReceiveMessage	MovementMode	RetrieveNeighboursList
JoinServer	LeaveServer	LeaveGame	JoinGame

Table 1. User/Service Interactionsn

3.3.2 Performance information elements

Performance metrics from lower layers are already supported by the IEEE 802.21; the ones being proposed relate to the application aware QoS and QoE values, which are directly reported from the application layer to the MIH User. The agent collects information from the several sensors in the vehicles as well as from the applications running. Vehicles become multimedia oriented as time evolves, and now include displays for video and gaming purposes, network connectivity for the retrieval of additional network based services, or even simple internet access by itself. The QoE values will require additional metrics that are relevant to characterize the service delivery quality.

QoS Performance Information Elements

As stated, it is required to have the complete view of the vehicles' performance in the management platform. Usually the metrics associated with the vehicles performance are related with the components of each type of vehicle. Different sensors are applied to the several parts and monitor temperature, pressure, speed, etc. Table 2 shows an example of Distributed Control Architecture using CAN (Johansson et al., 2005).

Powertrain and Chassis	Body electronics
Transmission control module	Driver information module
Engine control module	Steering wheel module
Brake control module	Rear, Frontal and Central modules
Door module	Climate control module
Steering angle sensor	Steering wheel module
Suspension module	Auxiliary electronic
Audio module	Infotainment control

Table 2. example from (Johansson et al., 2005) for a particular automotive integration solution

The defined information elements do not need to be gathered via the CAN solution. They can be collected via any customized solution as long as the agent receives them in the pre-defined

format and is able to send them to the network-side modules of management. For the purpose of integrating the evaluation of the vehicle's analysis in the cloud, we consider Information Elements to be of the type: TYPE_IE_VP - Type Information Element Vehicle Performance.

QoE Performance Information Elements

The QoE Performance Information Elements are related with the services being accessed by the users on the vehicle. The services can be bundled by the CSPs as part of the overall package and can be evaluated individually. These parameters were first described in Miguel Almeida (2009), where a more extensive study is performed, employing a 3GPP view while underlining the relevant parameters at each layer of a 3GPP network. In fact, these parameters are a first glance at the performance view of a service, and could be narrowed down, for problem identification purposes, until the lower layers. It defines the way in which the KPIs should be constructed and how they can be evaluated in a Cross Layer View, while in Igor Pais (2009), a more QoE centric analysis is performed. For each service we consider metrics such as the total setup waiting time for a service to be received (TYPE_IE_SP_WT), the Mean Opinion Score (TYPE_IE_SP_MOSQ), the Service Availability (TYPE_IE_SP_AVAILABILITY), the Lost Packets (TYPE_IE_SP_LOSS), the Time Resolution (TYPE_IE_SP_TIMERES for voice and TYPE_IE_SP_FPS for video), and, of course, the Bit Rate (TYPE_IE_SP_BR). All primitives include the following fields: SourceID (or application ID), Application Type ID, Time and Value.

3.4 Integrating with the cloud

In the previous sections we have been debating the collection mechanisms which allow to gather and convey the information from the vehicles into the network. This would allow the Mobile Network Operators (MNO) which own the gathering technology to access the data and evaluate it. In order to make it publicly available and, in this way, further capitalize this solution, the MNO would greatly benefit from a seamless way to make it available to 3rd party entities (e.g. 3rd Party CSPs), which could hire the access to the data. For instance, fleet management functions could be employed and then the vehicles' performance information might be outsourced. Subsection 2.2.1 shows a way to gather information using XMPP and then relaying it into the cloud using REST. By extending that proposal in order to also convey the information carried in the MIH messages, we create a compliant system. To do so, we need to create an adaptation in the Cloud Bridge Server (Miguel Almeida, 2010a), which we denote as Performance and Behavior API Agent. That agent interfaces with the MIH agents which collect the messages from the vehicles generating REST alike messages which update the 3rd party webservices accordingly. In this way, instead of using XMPP as the transport protocol, we use 802.21 to convey the performance messages, which is a more efficient solution for medium to high mobility scenarios (see Fig. 6). The web services expose an interface that allows information to be asynchronously supplied, and commands requested, when necessary, due to network management operations.

We gather the SAP ID and cross match it with the vehicle's information within the operator. There are two solutions to cope with the device identification. We couple the information of the IDs of the Service Access Points and of the MIH Users that coexist in the same vehicle and then translate them into a resource. By employing the translation shown in Fig. 7, we can guarantee the required consistency between the adaptation and the CBS, enabling the exposure of MIH resources to the REST interface, which can now be fully controlled on a

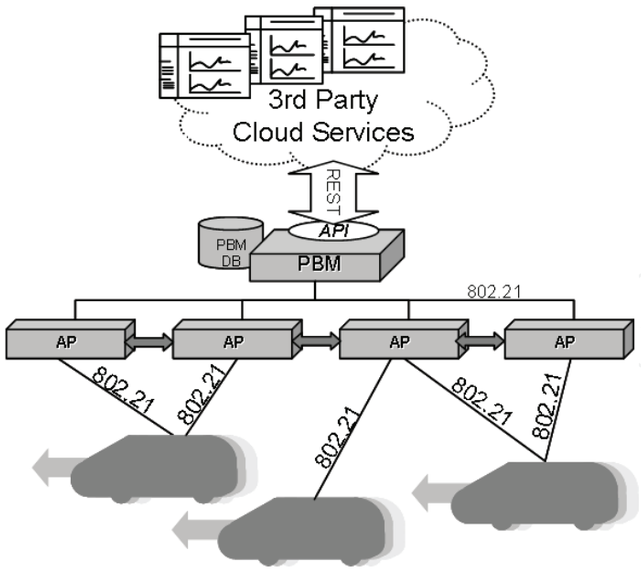


Fig. 6. Protocols Used for the interaction between the Vehicles and the Cloud

per-vehicle policy determined by the bridge. All messages are sent to the Cloud seamlessly, given that any web like environment can support RESTful primitives.

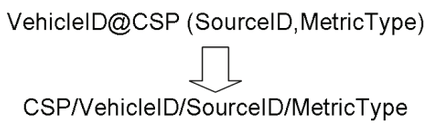


Fig. 7. Creating the ID for the REST resource

As the entry point towards the Cloud, the Performance and Behavior API Agent enables the communication between the devices and the Software as a Service, taking on a vital role for authentication and authorization. Each service must register, define an SLA, and authenticate to gain access to information pertaining to the vehicles. Given the control over the information, the Performance and Behavior API Agent is able to define a granular access control to the information exposed. The Performance and Behavior API Agent supports HTTP compliant messages (GET, PUT, UPDATE, DELETE), which are also the core of REST functionality, thus assuring the integration with minimal effort on the Cloud Bridge Server. This secure setup even allows customizing the devices towards a specific web service. If the devices are configured accordingly, they can opt to send reports to the custom Web Service URLs. The 802.21 signaling will transport the report, and then the Performance and Behavior API Agent will perform a HTTP POST on the destination Web Service. The Web Services need to be aware of the data model being communicated.

4. Performance comparison

The main advantage of the described approach is to save on signaling and overhead while simultaneously allowing end-to-end heterogeneous reports, and seamlessly integrating both reporting and network enforcement processes. The proposed architecture aims to provide an accurate profiling of users for an enhanced network optimization and resource consumption

prediction. In this section, it is presented a quantitative evaluation in terms of traffic generated by different approaches, and a qualitative evaluation in terms of supported functionalities.

4.1 Qualitative evaluation

In this section we include a qualitative evaluation of the functionalities provided by current approaches and the IEEE 802.21 based approach (denoted as MIHR). A summary of the supported features is presented in Table 3. We first start by comparing the different approaches in terms of features support. We can see that for the purposes of integrating the devices with a web environment, SNMP is the most inadequate, since HTTP based solutions are already web based, and XMPP is XML based which enables a direct transformation of the objects. The 802.21 approach requires a special adaptation on the network side. Since it only defines a way for the devices to intercommunicate with the network on a lower level, a MIH user is required at the network side to behave as a proxy towards the web cloud.

Regarding the security features, XMPP creates a secure unique channel using TLS. Earlier versions of SNMP present serious security constraints, and this has been addressed in SNMP v3; still, the common applications use IPSec bellow the SNMP communication. Although this feature does not reside within the main focus of the 802.21, it can use 802.1x for the IEEE based networks and use the channel security procedures of the 3GPP networks. SNMP also allows authentication to verify that the message is from a valid source, while XMPP with REST supports authentication of an Identity and its merging with accounting information. With respect to Identity Management, XMPP is the best proposal to link several devices to someone's identification. Considering that the platform is to be deployed at a Network Operator's site, then it would be good to allow the cross matching of accountings with the operator's database, only feasible using the possibilities offered by XMPP. Since we are focusing on the device management more than on the Identity management functions, it is simpler to use a lower layer device management system, which takes into account the SAP ID or simply an IMEI.

Feature:	MIHR	XMPP + REST	HTTP Based	SNMP
Security	Yes	TLS	SSL	IPSec
Reliability	Yes	Yes	No	Yes
Authentication	No	SASL	No	No
Web Cloud Integration	High	Easy	Easy	Medium
CSP Integration	Medium	Easy	N/A	Complex
Identity Management	No	Yes	No	No
Bi-Directionality	Yes	Yes	No	Yes*

Table 3. Management Features Comparison; *not when using traps

Being able to support asynchronous communication allows the deployment of a very relevant feature for fault management functions, which is sending alarms in a near real-time way. There are several applications that take advantage from this possibility, and XMPP is the best approach to deal which this type of events. Moreover, it allows bidirectional communication without the need to run a web server in the devices, which is the only way to support bidirectional communication using HTTP with REST or with SOAP. The way 802.21 handles this problem is through commands, which allow sending information to the devices. In this sense, by gifting the network with the appropriate intelligence at the MIH user, it is possible to enable bi-directionality of the management applications, in a very resource effective way.

IEEE 802.21 considers reliability to be a requirement from the underneath media, which needs to have a reliable message delivery procedure in order to allow the MIH Protocol Acknowledgment Service (802.21-2008, 2009). XMPP and HTTP run on top of TCP, which ensures the reliability mechanisms. SNMP runs on top of UDP, and therefore it employs the mechanism of request/response: if the response is not obtained, the request is again sent (and can be customized). Obviously, this a problem for the trap-enabled version of the protocol.
CDRs are call oriented and transport the information of the sessions and of the users involved. Their major advantage is the fact that they were created for an easy usage within the operator's domain. The MIHR requires an effort to integrate the solution with the Mobile Network Operator's (MNO) Home Location Register (HLR), while employing a mapping between the devices and the owners. For the integration of the HTTP based solutions, this would be difficult and would require a great deal of customizations, namely assuring that the agents collect such information on the client side, and then, convey it to operator. The degree of customization would be so high that we opt to define it as non applicable, since it is a concept too broad and too subjective. SNMP also would require an adaptation mechanism that is aware of the identification of the device with which the network is communicating. The network would then need to map this information with something previously known, i.e., it would need to previously be aware of the agent ID, and map it into a specific user.
A summary of the performance metrics is presented in Table 4. Using the IEEE 802.21 method, it is not required to exchange the full performance details, but only those which are required. Since we are dealing with a media independent proposal, our reports can be applied in any type of technology even in a switched domain. SNMP is technology oriented and MIBs are defined for specific types of hardware. Typically, CDR analysis is very technology-oriented and includes details relevant only for the source of the type of access. When using an approach on top of HTTP, it also becomes depends on the network information, and the typical approach is to perform QoS measurements and establish a TCP connection towards a collecting server. In IEEE 802.21, the MIIS already handles that interface seamlessly. XMPP uses TCP and runs on top of IP, so although it could be considered technology agnostic, it brings large amount of of signaling, from the presence stanzas. For cellular networks, this would require optimization procedures, and additionally, login/logout functionalities.

Metric	MIHR	XMPP + REST	CDR	HTTP Based	SNMP
Overhead	Good	Bad	Bad	Bad	Medium
Signaling	Good	Bad	Bad	Bad	Medium
Heterogeneity	Yes	Yes	Yes	Yes	Yes
Synchronous	Yes	Yes	Yes	Yes	via Traps
Asynchronous	Yes	Yes	Yes	Yes	Yes
Multilayer	Yes	Yes	No	N/A	No

Table 4. Qualitative Performance Comparison

Multilayer analysis relates to the ability of the different procedures to evaluate the performance at different layers, and in parallel, to deal with the end user behavior issues. Using the IEEE 802.21-based approach, information can be collected from lower layers and from upper layers, simultaneously or separately depending on which information is required. Information can be collected locally at the NEs and then be reported to a central server: an approach with reporting over IP will still work, but will be less seamless for the intermediate

NEs. The same concept applies to SNMP; however, usually this procedure will not be applied to upper layer analysis or behavior related parameters. On the other hand, CDRs will not focus on the network information. XMPP is dependent on what the agent is collecting, but it introduces no constraints at this level, relying only on the capabilities of the agent@terminal.
IEEE 802.21-based approach relies on the IEEE 802.21 mechanisms, which do not support events from upper layers. Having that in mind, it is only possible to support reactive events for some of the typical performance parameters. However, as explained before, we overcome this problem for the support of alarms via the issuing of customized messages from the terminals. Asynchronous events could be supported by implementing event triggering from upper layers for end user behavior analysis. Since event triggering from lower layers is still valid, this proposal partially supports synchronous reporting. XMPP was created for instant messaging purposes, so it excels at both approaches, while SNMP can only be considered to be asynchronous when using traps. CDRs are implemented on a per-call basis, so it can be considered an asynchronous approach. The HTTP based approach can also be considered proactive, since it requires inputs, but it supports both asynchronous and synchronous methods (e.g. Get method will request for information, while the Post will add a new resource). In order to employ a synchronous procedure with requests from the network to the client, this would require the devices to have a webserver installed, which is clearly a downside. When considering the heterogeneity support, one issue arises: assuming that most of the approaches can run on top of the IP/TCP stack, we would need to state that XMPP, HTTP and SNMP approaches that use TCP and UDP are heterogeneous as long as the technologies use IP. This is a fairly accepted assumption today. However, the only technology that was created having in mind heterogeneity support was the IEEE 802.21.
In terms of signaling, overhead and performance comparison, we evaluated how the access links that connect the devices to the cloud would behave. As can be seen in Section 4.2, SNMP outperforms the XMPP approaches. This happens mainly because in XMPP we defined an Object Class and used the objects within an XML to make the transactions. The results presented for the XMPP approach were obtained from experimental evaluation, while in the SNMP related results, we computed the overhead induced by raw data transportation, when conveying binary information. This information would require post processing at the network management system. The exchange of performance records in CDRs is typically fixed size oriented and depends on the record detail. Moreover, the overhead is significantly high, since it includes information related to the user device for identification (e.g. IMEI), which is not required with IEEE 802.21, since the SAP ID already matches the device ID. The major inconvenience of using CDRs with additional inputs from the terminals is that it requires additional overhead and signaling, since in 3GPP networks it is required to consider the tunneling effects and the establishment of IP or GPRS connections for data transmission. In the future 3GPP releases this problem may be mitigated; however, reporting at this level will always introduce an overhead larger than the IP+UDP one. The best approach is to use the IEEE 802.21 solution, which handles the problem on a lower layer basis, thus reducing overhead and signaling.

4.2 Quantitative evaluation

Figure 8 presents the traffic generated by the several approaches with the size of the object of information being reported. We compare the performance of different protocols on the wireless link, i.e., the link between the vehicles and the network's infrastructure nodes, which

is the most problematic part of the network, when considering high mobility scenarios. This is in fact the link which introduces more concerns, in terms of resource consumption efficiency. The evaluated protocols include: SNMP Polling or Trap-based methods, HTTP using SOAP (Simple Object Access Protocol), IEEE 802.21 and XMPP transporting REST methods. Regarding XMPP and REST, we used the results detailed in Miguel Almeida (2010a), which were experimentally obtained. The other metrics were obtained as detailed bellow. Typically, Mobile Web Services introduce high overheads in general. (M. Tian, 2003), (Pras et al., 2004) include details on the overhead impact under various conditions. As Figure 8 shows, when using an object based transport like the one used in MQA using SOAP over HTTP, one must consider the overhead introduced by the signaling and also the headers of the IP, TCP (with timestamps), SOAP and SOAP envelope. The calculated overhead is presented in Equation 1. As the object size increases, packet segmentation occurs at the TCP layer, thus significantly increasing the size of generated traffic.

$$length_{Soap} = header_{IP} + header_{TCP} + header_{HTTP} + +header_{Soap} + envelope + object_{size} \quad (1)$$

When using SNMP, the overhead per object is decreased for both scenarios: we consider both the best case scenario with unsolicited messages' exchange (via the usage of traps), and also in the case of polling-based approach. According to (de Lima et al., 2006), the header size of SNMPv2c is approximately 25 octets. The overhead of SNMPv3 is given by Equation 2, where the Header Data of the SNMP is given by Equation 3 as 17 octets. This means that SNMPv3 adds a minimum of 17 octets to SNMPv2c. Considering the signaling generated by both versions, the total generated traffic is given by Equation 4 and Equation 5, respectively for SNMPv2c and SNMPv3. In Figure 8, SNMPv3 traffic was generated using ASN.1 (Abstract Syntax Notation One), and Get processes (Request + Response messages) were taken into account for overhead measurement purposes. For the transmission of more objects, the performance of SNMP is decreased, as shown in (de Lima et al., 2006) and in Figure 9. Using a bulk procedure would greatly reduce the overhead in this case, but this would be true for all other proposals.

$$Traffic_{SNMPv3} = HeaderData + SecurityParameters + ScoopedPDUdata \quad (2)$$

$$Header_{SNMPv3} = Version + MSGID + MaxSize + Flags + SecurityModel = 17Octets \quad (3)$$

$$TrafficSNMPv2c = 54 + n(10 + 2.Objectlength + Value) \quad (4)$$

$$Traffic_{SNMPv3} = 88 + n(10 + 2.Object_{length} + Value) \quad (5)$$

When considering the trap-based approach, we determine the traffic of SNMP in Equation 6 for SNMPv2. Then, we calculate the minimum traffic generated by SNMPv2 traps in Equation 7 and add the additional minimum overhead (considering Communitysize=6 octets and Traplength=1 octet) to get Equation 8 for the traffic generated by SNMPv3 traps (OID refers to Object Identifier). As can be seen in Figure 8 and Figure 9, Traps reduce the generated traffic, especially when a large number of events are being generated.

$$Traffic_{TrapSNMPv2c} = 63 + Community_{size} + TrapOID_{size} + n.(3 + OID + Value) \quad (6)$$

$$Traffic_{SNMPv2Trap} = 70 + (3 + OID + Object_{size}).numObjects \quad (7)$$

$$Traffic_{SNMPv3Trap} = 87 + (3 + OID + Object_{size}).numObjects \quad (8)$$

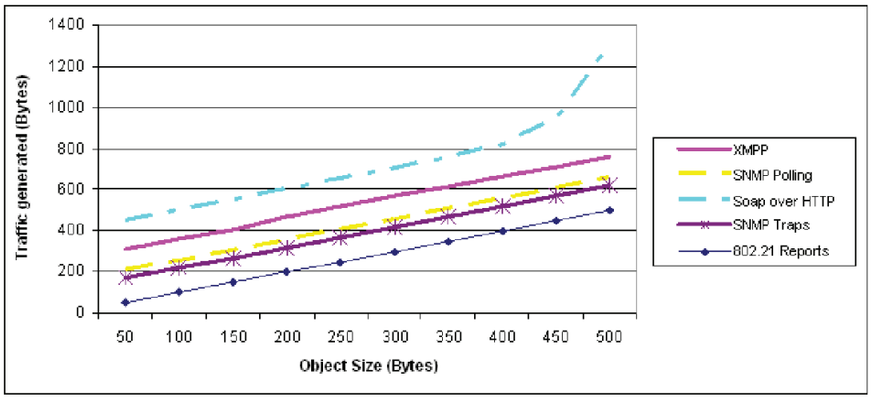


Fig. 8. Traffic Generated vs number of Objects

We also evaluated the minimum traffic generated by an approach which only uses UDP over IP without any additional signaling. Besides the application port, the OID and the value of the object, we only considered the values for an unsolicited procedure with the overhead of the IP and UDP headers and a variable field using one octet for the OID. The traffic generated consequently decreases when compared to SNMP. Considering our IEEE 802.21 approach, we can observe that it generates similar or lower traffic as an IP unicast transmission of the raw information over UDP. This becomes more clear in Figure 6, since MIH reports do not require the specification of a way to request for specific objects above the IP and UDP layer; the major overhead saving comes from the lack of requirement for the usage of the IP header and signaling, which would be required in order to support the request of an object and consequent transport. (Melia et al., 2007) further details this aspect with emphasis on the low overhead introduced by the protocol during mobility procedures.

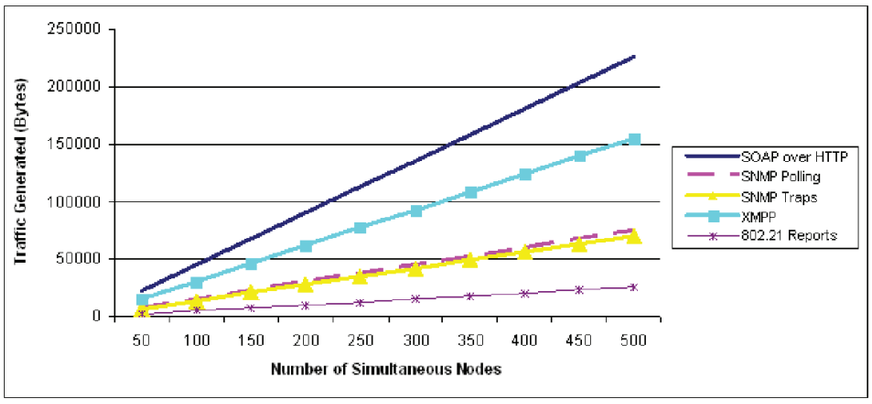


Fig. 9. Traffic generated by number of Nodes

5. Conclusions

In this chapter, we presented an architecture based on IEEE 802.21 that is able to gather information from vehicles and allows access via a web cloud. It provides seamless support at different levels: reporting of cross-layer information, support of

inter-technology environment, and integration of the actions of reporting with those of network reconfiguration. This approach combines the basic mechanisms of the IEEE 802.21 to gather information from the devices on the wireless link and the REST primitives to convey that information into the Cloud. Using the IEEE 802.21 Information Service, we underlined the required basic signaling to integrate both upper and lower layers information, regarding both the QoE the user is experiencing and the QoS parameters of the services. Moreover, using IEEE 802.21, the network can then act on the vehicles, basing its decision on profiling algorithms which estimate the future actions of a group of users, seamlessly supporting both mechanisms of reporting and reaction. The results presented show that this approach significantly decreases the reporting overhead, while it introduces a set of functionalities not present in current approaches.

Through the analysis of several transport technology (and as the core driver for interactions within the core domains), we consider a XMPP based solution to be the best approach when envisioning the integration of end users interacting with terminals, gaming consoles, cell phones, IP enabled sensors, etc, with web environments and also with the operator's infrastructure. However, for more mobile environments, where wireless resources are the major concern, and where fast connectivity maintenance procedures play a major role, its performance decreases. The number of features supported allows the authentication using the account identification of the devices' owners within the operator's Charging Gateways, allowing the easy deployment of charging per usage.

The integration with the cloud environment provided through REST interfaces allows the interaction with 3rd party web services, increasing the possibilities of applicability and revenue. By providing common and consistent interfaces to act and report on devices, we enable a new array of business relationships and opportunities that put the telecommunication and infrastructure operator back in the driver seat of the network, while enabling a clear interaction with the Cloud world, a feature which has been profoundly lacking from the operators portfolio. We believe that these paradigms will be a key revenue system where both operators and service providers can capitalize by using the adequate tools to unite the common approaches. This view greatly facilitates the interactions with vehicles on the move, via several technologies seamlessly reporting behavior and performance metrics using a lightweight reporting mechanism coupled with mobility

6. References

802.21-2008, I. S. (2009). Ieee standard for local and metropolitan area networks- part 21: Media independent handover, *IEEE Std 802.21-2008* pp. c1 –301.

Berners-Lee, T., F. R. & Masinter, L. (1998). *Uniform Resource Identifiers (URI): Generic Syntax*, IETF RFC 2396.
URL: *http://www.ietf.org/rfc/rfc2396.txt*

Breda, G. & Mendes, L. S. (2006). Qos monitoring and failure detection, *Telecommunications Symposium, 2006 International*, pp. 243 –248.

Chung, T.-Y., Yuan, F.-C., Chen, Y.-M., Liu, B.-J. & Hsu, C.-C. (2008). Extending always best connected paradigm for voice communications in next generation wireless network, *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pp. 803 –810.

de Lima, W., Alves, R., Vianna, R., Almeida, M., Tarouco, L. & Granville, L. (2006). Evaluating the performance of snmp and web services notifications, *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pp. 546 –556.

Don Box, David Ehnebuske, G. K. A. L. N. M. H. F. N. S. T. D. W. (2000). *Simple Object Access Protocol (SOAP) 1.1*, W3C Note.

Fabio Forno, P. S.-A. (2005). *XEP-0072: SOAP Over XMPP*, 1999 - 2010 XMPP Standards Foundation.
URL: *http://xmpp.org/extensions/xep-0072.html*

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*, Doctor of Philosophy Dissertation, University of California, Irvine.

Gurgen, L. & Honiden, S. (2009). Management of networked sensing devices, *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, pp. 502 –507.

Hornsby, A. & Bail, E. (2009). x03bc;xmpp: Lightweight implementation for low power operating system contiki, *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pp. 1 –5.

Hornsby, A., Belimpasakis, P. & Defee, I. (2009). Xmpp-based wireless sensor network and its integration into the extended home environment, *Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on*, pp. 794 –797.

Igor Pais, M. A. (2009). End user behavior and performance feedback for service analysis, *Intelligence in Next Generation Networks - ICIN*.

ITU (1996). *Telecommunications Management Network Reference Model: CCITT Recommendation M.3010*, International Telecommunication Union.

J. Case, E. A. (1990). *A Simple Network management Protocol (SNMP)*, RFC 1157.

Jesus, V., Sargento, S., Corujo, D., Senica, N., Almeida, M. & Aguiar, R. (2007). Mobility with qos support for multi-interface terminals: Combined user and network approach, *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pp. 325 –332.

Johansson, K. H., TÃűrngren, M. & Nielsen, L. (2005). *Vehicle Applications of Controller Area Network*, BirkhÃd'user.

Jung, S.-J., Lee, J.-H., Han, Y.-J., Kim, J.-H., Na, J.-C. & Chung, T.-M. (2007). Snmp-based integrated wire/wireless device management system, *Advanced Communication Technology, The 9th International Conference on*, Vol. 2, pp. 995 –998.

M. Tian, e. A. (2003). Performance considerations for mobile web services, *IEEE Communication Society Workshop on Applications and Services in Wireless Networks*, Vol. 2, pp. 741 –746.

Melia, T., de la Oliva, A., Vidal, A., Soto, I., Corujo, D. & Aguiar, R. (2007). Toward ip converged heterogeneous mobility: A network controlled approach, *Comput. Netw.* 51(17): 4849–4866.

Miguel Almeida, A. M. (2010a). Bridging the devices with the web cloud: A restful management architecture over xmpp, *6th International Mobile Multimedia Communications Conference*.

Miguel Almeida, Rui Inacio, S. S. (2009). Cross layer design approach for performance evaluation of multimedia contents, *International Workshop on Cross Layer Design*.

Miguel Almeida, S. S. (2010b). Media independent end user behavior and performance reports, *IEEE GLOBAL COMMUNICATIONS CONFERENCE*.

Mohinisudhan, G., Bhosale, S. & Chaudhari, B. (2006). Reliable on-board and remote vehicular network management for hybrid automobiles, *Electric and Hybrid Vehicles, 2006. ICEHV '06. IEEE Conference on*, pp. 1 –4.

P. Saint-Andre, E. (2004). *Extensible Messaging and Presence Protocol (XMPP): Core*, IETF RFC 3920.
URL: *http://www.ietf.org/rfc/rfc3920.txt*

P. Saint-Andre, E. (2008). *XEP-0238: XMPP Protocol Flows for Inter-Domain Federation*, 1999 - 2010 XMPP Standards Foundation.
URL: *http://xmpp.org/extensions/xep-0238.html*

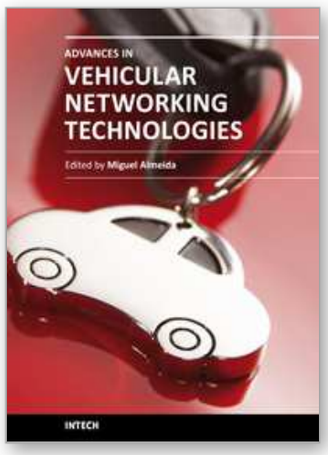
Peter Saint-Andre, Kevin Smith, R. T. (2009). *XMPP: The Definitive Guide Building Real-Time Applications with Jabber Technologies*, O'Reilly Media.

Pras, A., Drevers, T., van de Meent, R. & Quartel, D. (2004). Comparing the performance of snmp and web services-based management, *Network and Service Management, IEEE Transactions on* 1(2): 72 –82.

Soldani, D. (2006). Means and methods for collecting and analyzing qoe measurements in wireless networks, *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, pp. 5 pp. –535.

Voas, J. & Zhang, J. (2009). Cloud computing: New wine or just a new bottle?, *IT Professional* 11: 15–17.

Waldbusser, S. (1995). *Remote network Monitoring management Information Base*, RFC 1757.



Advances in Vehicular Networking Technologies

Edited by Dr Miguel Almeida

ISBN 978-953-307-241-8
Hard cover, 432 pages
Publisher InTech
Published online 11, April, 2011
Published in print edition April, 2011

This book provides an insight on both the challenges and the technological solutions of several approaches, which allow connecting vehicles between each other and with the network. It underlines the trends on networking capabilities and their issues, further focusing on the MAC and Physical layer challenges. Ranging from the advances on radio access technologies to intelligent mechanisms deployed to enhance cooperative communications, cognitive radio and multiple antenna systems have been given particular highlight.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Miguel Almeida and Susana Sargento (2011). Coupling Activity and Performance Management with Mobility in Vehicular Networks, Advances in Vehicular Networking Technologies, Dr Miguel Almeida (Ed.), ISBN: 978-953-307-241-8, InTech, Available from: http://www.intechopen.com/books/advances-in-vehicular-networking-technologies/coupling-activity-and-performance-management-with-mobility-in-vehicular-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen